

基于 H.264 编码帧内预测模式的视频监控算法研究

张娜,周冬,张俊为,桂江生,包晓安

(浙江理工大学信息学院,杭州 310018)

摘要: 为了降低视频监控中 H.264 编码帧内预测模式计算复杂度,在分析了帧内预测模式选择过程的基础上,提出了一种改进的提前终止选择算法。该算法根据画面的抖动状况以及 INTRA16×16 模式下的纹理特征来获取最佳阈值,并根据预处理阈值计算出最佳预测模式,从而降低了预测模式的计算复杂度,增强了在不同监控场景下的适应能力。将该算法与模式选择算法、多级模式预测方法进行实验分析,分别在比特率、帧预测次数和计算时间上进行比较,结果表明:通过改进后的算法可以有效去除模式的预测数,避免了不必要的预测计算,同时保持了 H.264 编码原有的特性。

关键词: 视频监控;视频传输;H.264 编码;帧预测模型

中图分类号: F252

文献标志码: A

文章编号: 1673-3851(2017)03-0432-06

0 引言

随着视频技术越来越多地应用在各行各业中,视频的编码处理显得尤为重要。一般的视频监控摄像头所采集到的数据量比较大,当遇到网络带宽比较低的情况下,将未经过编码压缩处理的数据直接在网络上进行传输,不仅会增加带宽的压力,而且对摄像头硬件也会造成影响^[1]。视频编码是利用视频编码数据宏块之间的相关性,通过建立视频编码模型以去除冗余数据的过程^[2]。经过编码可大幅减少传输数据量。目前常用的视频编码技术有 H.264、MPEG 等^[3]。其中 H.264 数据编码具有高压缩比和网络传输性,在同等视频质量的情况下编码的码流只有其它编码方法的一半,但也存在编码计算复杂等弊端。

帧内预测是 H.264 编码中的主要技术,其实现原理是利用图像中的空间相关性,由已经解码的相邻块来预测当前块的信息^[4]。H.264 编码方法在选择帧内预测模式时采用全搜索技术,通过计算少量的数据来获取好的编码效果,但由于 H.264 的

帧内预测模式数目很多,为计算出某一宏块的帧内预测模式,需要分析并比较 592 种不同组合模式下的代价函数,这使得编码复杂度非常高^[5]。但对于实际视频监控的应用场景,监控摄像头位置固定不变,监控画面一般没有较复杂的跳变,大多都是处于平缓状态,编码过程中很多模式预测计算是多余的,因此选择快速准确预测帧内模式非常重要^[6]。

目前,有关帧内预测方法的改进研究主要有快速帧内预测算法,如: Huang 等^[7]针对边缘方向预处理提出了一种模式选择算法;杨秀增^[8]提出了利用绝对差和方向信息的多级模式预测方法。这两种方法都是采用 RDO(rate-distortion optimization)进行模式选择。RDO 计算复杂度较高,当监控画面分辨率高时,这两种算法的时间性能比较差。本文根据帧内预测模式的选择过程,对传统预测模式与 INTRA4×4 模式下的预测算法进行改进,提出了一种简单有效的预测模式选择方法,改进后的算法能预先减少 INTRA16×16 模式选择,有效降低模式选择的计算复杂度,同时基本保持了 H.264 原有的性能,符合了视频监控的需求。

收稿日期: 2016-11-24 网络出版日期: 2017-03-28

基金项目: 国家自然科学基金项目(61379036,61502430);浙江省自然科学基金项目(LY12F02041);浙江省重大科技专项重点工业项目(2014C01047)

作者简介: 张娜(1977-),女,浙江杭州人,副教授,硕士,主要从事自适应软件、软件测试与智能信息处理方面的研究。

通信作者: 包晓安, E-mail: baobaoan@zstu.edu.cn

1 H.264 帧内预测模式

在 H.264 编码结构中,以宏块作为基本单位进行视频编码。在帧内编码过程中,将每个宏块划分为一个 16×16 亮度像素块和两个 8×8 彩色像素块,并分别对这 3 个块进行帧内预测。亮度块又可分为 INTRA4×4 和 INTRA16×16 两种模式帧内预测类型^[9]。其中 INTRA4×4 模式有 9 种帧内预测类型,INTRA16×16 模式有 4 种帧内预测模式,后者如表 1 所示^[10]。

表 1 INTRA16×16 的帧内预测模式

预测模式	预测方式
预测模式 0	采用垂直方向,从上边像素来推算当前像素值
预测模式 1	采用水平方向,从左边像素来推算当前像素值
预测模式 2	采用 DC 预测,从左边和上边的像素来推算当前像素值
预测模式 3	利用 plane 函数及左边上边像素推算当前像素值

INTRA4×4 模式通过对 4×4 子块进行预测分析,其预测精度较高,所以更适合用于处理画面细节多、纹理复杂的视频场景,如道路、车站、广场等场景^[11];INTRA16×16 模式通过对 16×16 子块进行预测,预测方向如图 1 所示,预测精度较低,适用于画面变换平缓的监控场景,如固定视角的家庭、办公监控场景^[12]。本文主要对 INTRA16×16 模式的选择过程进行相应改进。

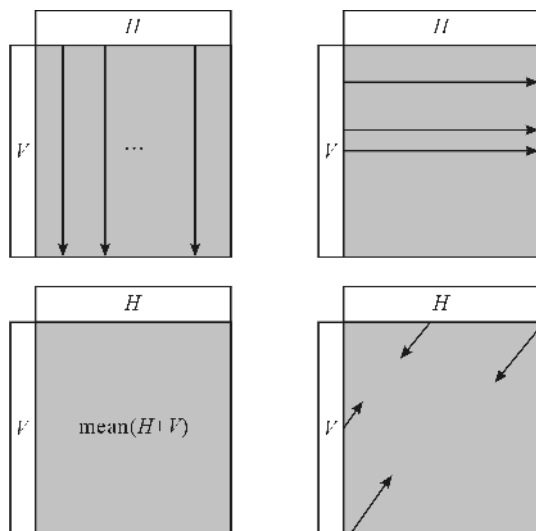


图 1 INTRA16×16 的帧内预测模式图示

2 帧内预测的分析

2.1 相邻块之间的相关性

为计算当前数据块,除需获取所用的预测模式外还需计算相邻块之间的差值。如果用一个值标记每个编码块的预测模式,这样需要消耗较大的编码比特率,将会增加码率^[13]。事实上 INTRA16×16 模式下相邻块之间的预测模式具有很强的相关性,可以据此计算最有可能的模式,减少编码比特数。

2.2 帧内预测模式选择过程

计算出某个宏块内的帧预测模式,需要计算 592 种代价函数,计算量非常大。He 等^[12]提出的计算方法,首先计算 9 种 INTRA4×4 模式代价函数,得到最小代价的模式。计算模式的代价函数 RD_{cost} 定义为:

$$RD_{cost} = SSD(N) + R_{mode} \times rate \quad (1)$$

其中: $SSD(N)$ 是编码模式 N 的失真度^[12],该值是编码块原始采样值 $s(i,j)$ 和预测样点值 $p(i,j)$ 之间的差值 $d(i,j)$ 绝对值之和,即:

$$SSD(N) = \sum_{i,j} |s(i,j) - p(i,j)| \quad (2)$$

$rate$ 表示熵编码后码率; R_{mode} 按式(3)计算:

$$R_{mode} = 0.85 \times 2(QP - 12)/3 \quad (3)$$

QP 表示量化参数。然后,通过式(1)算出 9 种 INTRA4×4 块最小的 RD_{cost} ,从而得到当前宏块 INTRA4×4 模式下的 RD_{cost} 。再通过上述方法计算出 4 个 16×16 块最小的 RD_{cost} ,比较前两步获得的 RD_{cost} ,计算出最小 RD_{cost} 的模式来作为宏块的帧内预测模式。通过这种方式计算宏块的最佳模式,复杂度相当大,而且对于相对平稳的监控画面会存在大量的数据冗余。

3 改进的预处理阈值提前终止选择算法

降低帧预测复杂度一般有两种方法:一种是通过减少代价函数 RD_{cost} 的计算量;另一种是减小帧预测模式的次数。本文在分析代价函数的计算量与帧预测模式次数的基础上,结合固定场所视频监控应用系统应用场景,减小帧预测模式的次数。成运等^[9]研究发现,通过当前编码块与周围相邻块之间的相关特征预先排除可能性小的预测模式,可以降低帧内预测的复杂度。本文在原有的帧预测模式的基础上,提出了一种基于改进预处理阈值提前终止模式选择算法,利用空间域的特征改变了 RDO 帧内预测

模式,在预测之前去除一些可能性低的预测模式,减少计算量,降低复杂度。

传统帧内预测算法^[13]是计算 $\text{INTRA}4 \times 4$ 与 $\text{INTRA}16 \times 16$ 下所有模式的率失真值,然后再通过计算比较得出最小代价的模式,这种方法的计算量

非常大。监控画面的图像都会存在一定比例的平缓区域,在本文提出改进算法中,预先设定经验阈值 T , T 按照传统帧内预测算法从 $\text{INTRA}16 \times 16$ 模式中最佳帧内预测模式的最小率失真值 RD_{cost} 加权平均值得出。算法的具体流程如图2所示。

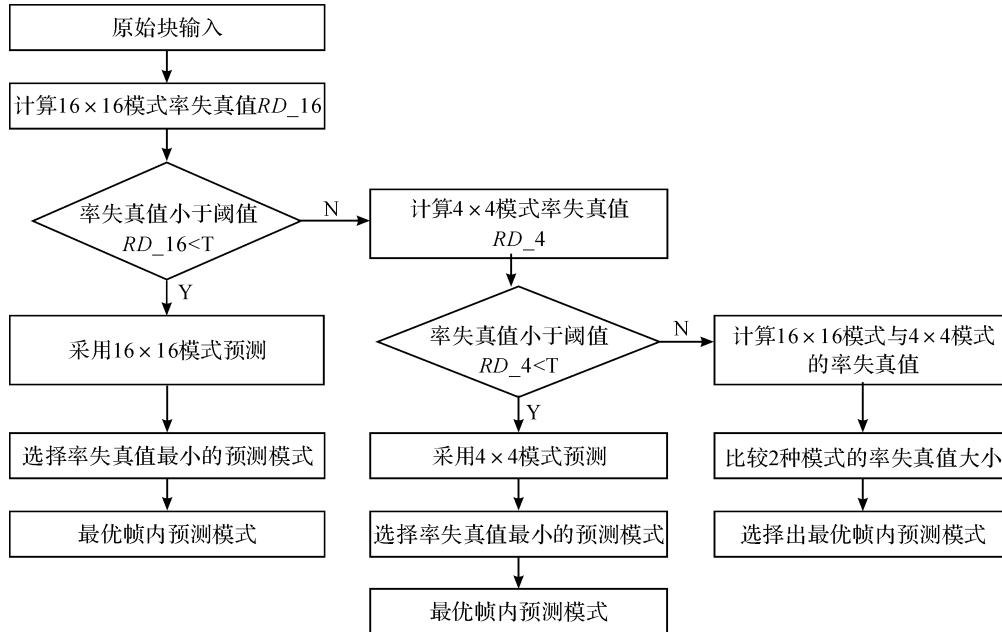


图2 改进算法流程

本方法可总结为如下4个步骤:

a) 首先算出4种 $\text{INTRA}16 \times 16$ 模式的代价函数,选择最小代价的模式。代价函数计算 $RD_{\text{cost}} = SSD(N) + R_{\text{mode}} \times \text{rate}$ 。

b) 比较 $\text{INTRA}16 \times 16$ 模式下的 RD_{cost} 与 T 之间的大小,若 RD_{cost} 比 T 小,则终止计算,选择最小的模式;否则转到步骤(c)。

c) 若 $\text{INTRA}16 \times 16$ 模式下的 RD_{cost} 比 T 大,则计算 $\text{INTRA}4 \times 4$ 模式下的 RD_{cost} ,比较 T 与 $\text{INTRA}4 \times 4$ 模式下的 RD_{cost} 的大小,若比 T 小,则终止计算;否则转到步骤(d)。

d) 若 $\text{INTRA}4 \times 4$ 模式下的 RD_{cost} 比 T 大,则比较 $\text{INTRA}16 \times 16$ 模式下的 RD_{cost} 与 $\text{INTRA}4 \times 4$ 模式下的 RD_{cost} 的大小,选择最佳的模式计算。

结合上述步骤和图2中的流程图,给出本文算法的实现方式如下:

1. Begin;
2. for $m = 1:4$
3. count $SSD_{16 \times 16}$ // 循环计算4种 $\text{INTRA}16 \times 16$ 模式的失真度
4. count $R_{16 \times 16}$
5. count $RD_{16 \times 16} = SSD_{16 \times 16} + r_m R_{16 \times 16}$ // 计算

$\text{INTRA}16 \times 16$ 模式下代价函数

6. if($RD_{16 \times 16} < T$)

7. choice best model_{16x16}

8. end

9. else if

10. for $m = 1:9$

11. count $SSD_{4 \times 4}$ // 循环计算9种 $\text{INTRA}4 \times 4$ 模式的失真度

12. count $R_{4 \times 4}$

13. count $RD_{4 \times 4} = SSD_{4 \times 4} + r_m R_{4 \times 4}$ // 计算

$\text{INTRA}4 \times 4$ 模式下代价函数

14. if($RD_{4 \times 4} < T$)

15. choice best model_{4x4}

16. end

17. else compare ($RD_{4 \times 4}, RD_{16 \times 16}$) // 比较两种模式下的代价函数

18. choice best model

19. end

20. End

在本算法中通过设置阈值 T 来判断出最佳模式,不但可以减少 $\text{INTRA}16 \times 16$ 模式下帧预测的计算量,在某些情况下甚至可以直接避免计算 $\text{INTRA}4 \times 4$

4 模式,在一定程度上提高了预测的精度。

在真实的视频监控场景中,需要考虑画面出现跳变或画面密集程度较高的情况。在这种情况下,如果采用阈值 T 来预测跳变的视频监控场景会有所误差。当阈值 T 预测失效时,通过计算 INTRA 16×16 纹理特征来选择出块的最佳预测模式。纹理特征提取是利用图像的侦测技术来检测纹理基元,建立出模型来获得纹理定量及定性描述的过程。因为简单的纹理区像素值分布相对集中,当画面出现跳变时可以计算出纹理复杂度 Tex 来预测帧的最佳模式,INTRA 16×16 模式下 Tex 是由各块像素值与其平均值之差的均值算出。由于 INTRA 16×16 有 4 种不同的预测方向,若纹理特征在预测方向时采用这 4 种预测模式,分别定义垂直预测 Tex_0 、水平预测 Tex_1 、DC 预测 Tex_2 及 $plane$ 预测 Tex_3 ,其纹理复杂度计算如式(4)~式(7)所示:

$$Tex_0 = \sum_{i=1}^{16} \frac{1}{16} \sum_{j=1}^{16} \left| p(i,j) - \frac{1}{16} \sum_{j=1}^{16} p(i,j) \right| \quad (4)$$

$$Tex_1 = \sum_{j=1}^{16} \frac{1}{16} \sum_{i=1}^{16} \left| p(i,j) - \frac{1}{16} \sum_{i=1}^{16} p(i,j) \right| \quad (5)$$

表2 算法测试结果

方法	参数	Q=24	Q=26	Q=28	Q=30
模式选择算法	比特率/kbs	1378.2	1355.9	1343.7	1302.5
	计算时间/ms	796.4	832.1	802.4	823.2
	预测次数	15315	15489	15407	15121
多级模式预测算法	比特率/kbs	1352.5	1342.5	1321.6	1292.4
	计算时间/ms	880.2	870.2	865.5	850.7
	预测次数	15279	15424	15390	15016
本文算法	比特率/kbs	1230.14	1225.15	1217.23	1210.16
	计算时间/ms	450.0	443.6	430.3	421.4
	预测次数	15245	15324	15289	14890

从表2可以看出,多级模式预测算法相较于模式选择算法,具有较低的比特率和帧预测次数,但需要消耗更多的时间。而本文提出的改进算法,在均占有绝对的优势,在比特率平均与平均帧内预测次数都有了很大程度的减少,比特率平均减少了1.65%,INTRA 16×16 模式下帧预测次数减少了约55%,有效的减少了代价函数的计算。

为了更加全面地对比3种算法,从峰值信噪比上进行比较,结果如图3所示。从图3中可以得出使用改进后算法得到的编码结果与模式选择算法和多级模式预测算法方法相比,峰值信噪比平均减少了2.05dB。

$$Tex_2 = \frac{1}{16} \sum_{i=1}^{16} \sum_{j=1}^{16} \left| p(i,j) - \frac{1}{16} \sum_{i=1}^{16} \sum_{j=1}^{16} p(i,j) \right| \quad (6)$$

$$Tex_3 = plane \sum_{i=1}^{16} \sum_{j=1}^{16} \left| p(i,j) - \frac{1}{16} \sum_{j=1}^{16} p(i,j) \right| \quad (7)$$

选择其中最小值 Tex_{\min} 作为 INTRA 16×16 块的纹理复杂度的衡量标准并选择出最佳模式:

$$Tex_{\min} = \min\{Tex_0, Tex_1, Tex_2, Tex_3\}$$

当画面出现跳变或画面密集程度较高的情况下,T值无法预测出最佳模式时采用纹理特征的复杂度来选择最佳模式具有很好的适应性。

4 实验及结果分析

为了验证本文提出的改进预处理阈值提前终止选择算法的有效性,将本文算法与模式选择算法^[7]、多级模式预测方法^[8]进行实验对比,编码使用 foremanQCIF 序列,分别在比特率、帧预测次数和算法计算时间上进行比较,得到3种算法的对比结果如表2所示。

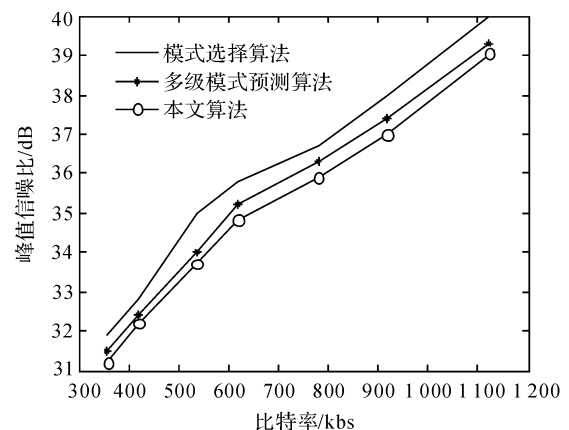


图3 三种算法峰值信噪比比较

同时将本文改进的算法应用到视频监控系统中,通过对摄像头里产生的日志编码数据来分析(如图4),可得出在相同的帧序列中,利用本文的算法可以产生比较少的字节数据,有效的减缓了摄像头的监控压力。

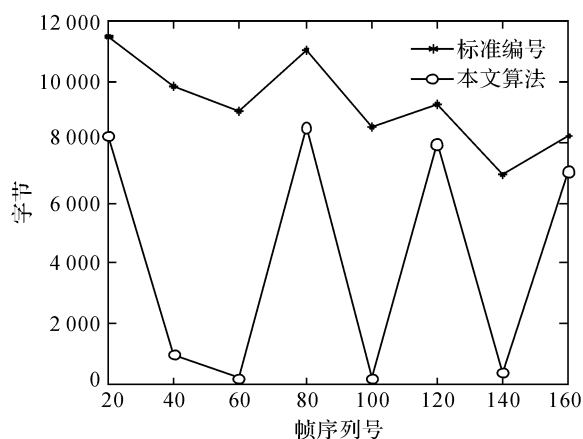


图4 本文算法与标准编码字节比较

5 结 语

本文根据 H. 264 编码的帧内预测过程以及编码块之间的相关性与帧预测模式间的特点,提出了一种改进的预处理阈值提前终止模式选择算法。该算法通过提前设置阈值来减少不必要的计算,从而减少了预测复杂度,并基本上保持了 H. 264 原有的性能。本文算法与标准编码视频监控中产生的字节进行对比,发现本文算法更适用于视频监控。后续研究将根据 H. 264 模式之间的相关性,进一步降低帧预测的代价计算,提升视频监控场景的图像传输的质量。

参考文献:

- [1] 黄凯奇,陈晓棠,康运锋,等. 智能视频监控技术综述[J]. 计算机学报,2015,38(6):1093-1118.
- [2] 钟锋,陆以勤. 基于家庭网关的嵌入式远程图像监控系统[J]. 计算机工程与设计,2011,32(5):1626-1629.
- [3] CHAABOUNI A, GAUDEAU Y, LAMBERT J, et al.

Subjective and objective quality assessment for H264 compressed medical video sequences[C]// International Conference on Image Processing Theory, Tools and Applications. IEEE,2014:1-5.

- [4] 裴世保,李厚强,俞能海. H. 264/AVC 帧内预测模式选择算法研究[J]. 计算机应用,2005,25(8):1808-1810.
- [5] SONG R, WANG Y L, HAN Y, et al. Statistically uniform intra-block refresh algorithm for very low delay video communication [J]. Frontiers of Information Technology & Electronic Engineering, 2013, 14 (5): 374-382.
- [6] 张小红,张媛. H. 264/AVC 帧内预测模式快速选择算法研究[J]. 系统仿真学报,2013,25(12):3050-3056.
- [7] HUANG Y H, OU T S, CHEN H H. Fast decision of block size, prediction mode, and intra block for H. 264 intra prediction [J]. Circuits & Systems for Video Technology IEEE Transactions on, 2010, 20 (8): 1122-1132.
- [8] 杨秀增. 基于 FPGA 的图像预测编码压缩算法的实现[J]. 电子技术,2013,40(2):62-63.
- [9] 成运,戴葵,王志英,等. 一种新的基于 H. 264/AVC 的零块判决方法[J]. 计算机研究与发展,2005,42(10): 1758-1762.
- [10] LIU B, CHEN Q. Implementation and optimization of intra prediction in H264 video parallel decoder on CUDA[C]// IEEE Fifth International Conference on Advanced Computational Intelligence. IEEE, 2012: 215-232.
- [11] 张剑,谭劲. 基于结构模块化实现 H. 264 编码软件[J]. 计算机科学,2005,32(6):87-90.
- [12] HE Z, CHEN C, CAI J. Lowpass filtering of rate-distortion functions for quality smoothing for real-time video recording and streaming [C] // International Conference on Image Processing. IEEE, 2004: 1101-1104.
- [13] 王刚,毛剑飞,田青,等. 基于 ARM11 的无线视频监控系

Research on Video Surveillance Algorithm Based on Intra Prediction Mode of H.264 Coding

ZHANG Na, ZHOU Dong, ZHANG Junwei, GUI Jiangsheng, BAO Xiaolan

(School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: In order to reduce the computational complexity of intra prediction mode of H.264 coding in video surveillance, an improved early termination selection algorithm is proposed based on the analysis of the intra prediction mode selection process. The algorithm obtains the optimal threshold value according to the jitter condition of the picture and the texture feature in the INTRA 16×16 mode, and calculates the optimal prediction mode according to the preprocessing threshold value, which reduces the computational complexity of the prediction mode and enhances the adaptability under different surveillance scenes. The algorithm is compared with the mode selection algorithm and the multi-level mode prediction method in terms of bit rate, times of frame prediction and computation time. The simulation results show that the improved algorithm can effectively remove the prediction number of the pattern and avoid unnecessary prediction calculation, while maintaining the original characteristics of H.264 coding.

Key words: video surveillance; video transmission; H.264 coding; frame prediction model

(责任编辑: 康 锋)